

Chapter Five: Contents**(Microsimulation – 10 December 2002 – LA-UR 01-5712 – Portland Study Reports)**

1. INTRODUCTION.....	1
2. PARAMETER SETTINGS.....	2
3. APPLICATIONS TO PORTLAND STUDY	5
3.1 PROSPECTS	6
4. GENERIC SIGNAL TUTORIAL.....	7
4.1 GENERIC SIGNAL BUILDER	7
4.2 ASSESSING GENERIC SIGNAL RESPONSE	9
4.3 TROUBLESHOOTING	12

Chapter Five—Microsimulation

1. INTRODUCTION

This section documents the input parameters and conditions that were set to generate travel times from plan files using the Traffic Microsimulator. The general documentation for the Traffic Microsimulator may be obtained from the TRANSIMS web site <http://transims.tsasa.lanl.gov>. A summary of that document is given here.

The detailed daily plans for each person in the population, including truck drivers and other transients, are generated by the Route Planner as described in the previous chapter. The Traffic Microsimulator executes all of these plans simultaneously on the transportation infrastructure, taking into account interactions between vehicles. In particular, the Traffic Microsimulator models realistically such congestion effects as:

- traffic jams
- difficulty making left turns and lane changes
- mass transit vehicles running behind schedule
- mass transit vehicles full
- waiting through multiple cycles of traffic signals

Moreover, these effects not only happen at the locations and times where the demand/capacity ratio is high, but they also spill back into upstream traffic and later times.

2. PARAMETER SETTINGS

There are many parameters controlling the execution of the Traffic Microsimulator, but none are specific to a particular city. The parameters important for modeling traffic flow have been set based on calibration runs on special-purpose networks. These parameters control lane-changing and other aspects of the basic vehicle-following driving model. They were not tuned to Portland traffic, but to the Highway Capacity Manual. As noted in Volume Four (*General Results (not validation)*), Chapter Four (*Microsimulation*), we adjusted some parameters and network tables to address issues that arose during the feedback process.

Most of the parameters specific to the Portland Case Study were thus input/output settings. For a more detailed description of the meanings of these parameters, see the Output subsystem documentation. Input parameters specify the location of the network tables, plan files, and vehicle files. Output parameters specify what to include in snapshot, summary, and event output. Some of these, such as snapshot data, are optional; others must be set to ensure that information required in the feedback processes is collected.

Among the feedback data are summary density data for every link in the network, specified by the following configuration file keys:

OUT_SUMMARY_NAME_8	summary.dens
OUT_SUMMARY_TYPE_8	TIME; DENSITY
OUT_SUMMARY_BEGIN_TIME_8	0
OUT_SUMMARY_END_TIME_8	86400
OUT_SUMMARY_TIME_STEP_8	900
OUT_SUMMARY_SAMPLE_TIME_8	60
OUT_SUMMARY_BOX_LENGTH_8	97.5
OUT_SUMMARY_FILTER_8	COUNT>0

This specification directs the Traffic Microsimulator to collect data on the density and travel times of traffic every 60 seconds, and to print out a summary of that data once every 15 minutes.

Only those links for which at least one vehicle left the link during the 15 minutes will be included in the output. The number of vehicles leaving a link is accumulated over the 60-second sampling period, so no vehicles are missed. The data from this output file will be used directly by the Route Planner to determine link travel times.

There is no way to distinguish between links with no vehicles present and completely jammed links in the data above, so we add another file with the following specifications:

OUT_SUMMARY_NAME_11	summary.noflow.dens
OUT_SUMMARY_TYPE_11	TIME; DENSITY
OUT_SUMMARY_BEGIN_TIME_11	0
OUT_SUMMARY_END_TIME_11	86400
OUT_SUMMARY_TIME_STEP_11	900

```

OUT_SUMMARY_SAMPLE_TIME_11      60
OUT_SUMMARY_BOX_LENGTH_11       97.5
OUT_SUMMARY_FILTER_11           COUNT==0 ; VCOUNT>0

```

These are links that had vehicles present, but did not have vehicles leaving. The travel time for these links will be reported as 0, but we can adjust that to any desired value before feeding it back to the Route Planner.

We also collect data for comparison with traffic counts along cutlines supplied by Portland. The output specifications are:

```

OUT_SUMMARY_NAME_9              cutline.auto
OUT_SUMMARY_TYPE_9              TIME
OUT_SUMMARY_BEGIN_TIME_9        0
OUT_SUMMARY_END_TIME_9          86400
OUT_SUMMARY_TIME_STEP_9         900
OUT_SUMMARY_SAMPLE_TIME_9       15
OUT_SUMMARY_BOX_LENGTH_9        97.5
OUT_SUMMARY_LINKS_9             $TRANSIMS_ROOT/data/cutlines-2001-03-22.txt
OUT_SUMMARY_VEHICLE_TYPE_9      AUTO
OUT_SUMMARY_FILTER_9            COUNT>0
OUT_SUMMARY_SUPPRESS_9          LANE ; SUM ; VSUM ; SUMSQUARES ; VSUMSQUARES ; TURN ; VCOUNT

```

```

OUT_SUMMARY_NAME_10             cutline.truck
OUT_SUMMARY_TYPE_10             TIME
OUT_SUMMARY_BEGIN_TIME_10       0
OUT_SUMMARY_END_TIME_10         86400
OUT_SUMMARY_TIME_STEP_10        900
OUT_SUMMARY_SAMPLE_TIME_10      15
OUT_SUMMARY_BOX_LENGTH_10       97.5
OUT_SUMMARY_LINKS_10            $TRANSIMS_ROOT/data/cutlines-2001-03-22.txt
OUT_SUMMARY_VEHICLE_TYPE_10     TRUCK
OUT_SUMMARY_FILTER_10           COUNT>0
OUT_SUMMARY_SUPPRESS_10         LANE ; SUM ; VSUM ; SUMSQUARES ; VSUMSQUARES ; TURN ; VCOUNT

```

These collect the number of autos (in the *cutline.auto* file) and trucks (in the *cutline.truck* file), leaving the link for only those links listed in the *cutlines-2001-03-22.txt* file. The data is summarized in 15-minute intervals. Because we are interested only in traffic counts for this data, we filter out most other columns.

Establishing cutline data for mass transit must be handled slightly differently because the Traffic Microsimulator does not distinguish a mass transit vehicle type. The following configuration file keys accomplish what is required:

```

OUT_EVENT_NAME_7                cutline.transit
OUT_EVENT_TYPE_7                TRAVELER
OUT_EVENT_SUPPRESS_7             ACCELS ; ANOMALY ; DISTANCESUM ; LEG ; LOCATION ; SIGNALS ; STATUS ; STOPS ; STOPPED ; TIMESUM ; TRIP ; TURN ; USER ; VSUBTYPE ; YIELDS
OUT_EVENT_FILTER_7              STATUS&2 ; STATUS&1 ; ROUTE>0
OUT_EVENT_BEGIN_TIME_7          0
OUT_EVENT_END_TIME_7            86402
OUT_EVENT_LINKS_7               $TRANSIMS_ROOT/data/cutlines-2001-03-22.txt

```

The status flag filters mean “the vehicle is on a link” and “the vehicle’s on-link status has just changed.” Taken together, these select only vehicles just entering a link. The requirement `ROUTE>0` selects only mass transit vehicles.

The mode feedback process requires knowledge of which travelers completed their trips and how long it took them. This is contained in a file specified by the following configuration file keys:

```
OUT_EVENT_NAME_3          endtrip
OUT_EVENT_TYPE_3          TRAVELER
OUT_EVENT_SUPPRESS_3      ACCELS;LEG;ROUTE;SIGNALS;STOPS;STOPPED;TURN;USER;VEHICLE;VEHTYPE;VSTYPE;YIELDS
OUT_EVENT_FILTER_3        STATUS&8; STATUS!&4
OUT_EVENT_BEGIN_TIME_3    0
OUT_EVENT_END_TIME_3      86402
```

The status flags used in the filter mean “the on-leg status has changed” and “the traveler is not currently executing any leg of a plan.” This is equivalent to “the traveler has just finished a trip.”

Among the optional data we track are reports of anomalies from travelers. We lump these into two files: one for travelers who become off-plan because of unanticipated congestion; another for most other anomalies. Other anomalies include events such as not being able to board a transit vehicle because it is full. A complete table is given in the documentation. We have not included anomaly 0, which is generated each time a vehicle cannot make a desired lane change because of congestion.

```
OUT_EVENT_NAME_4          anomaly.offplan
OUT_EVENT_TYPE_4          TRAVELER
OUT_EVENT_SUPPRESS_4      ACCELS;ROUTE;SIGNALS;STOPS;TURN;USER;VEHTYPE;VSTYPE;YIELDS
OUT_EVENT_FILTER_4        STATUS&8388608; ANOMALY==1
OUT_EVENT_BEGIN_TIME_4    0
OUT_EVENT_END_TIME_4      86402

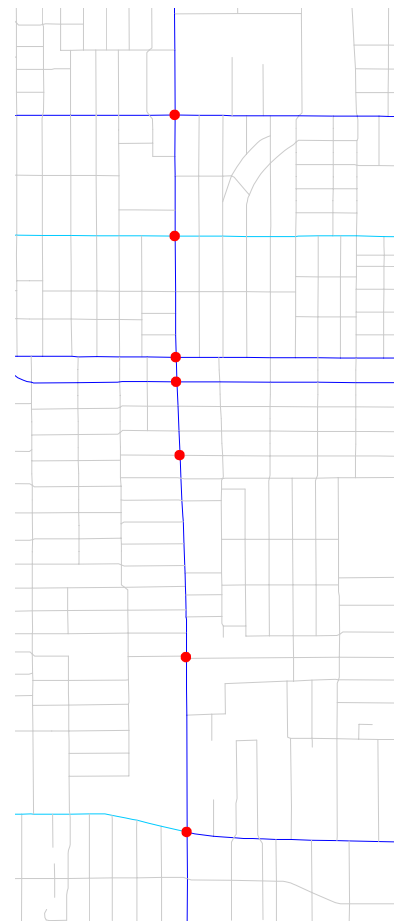
OUT_EVENT_NAME_5          anomaly.other
OUT_EVENT_TYPE_5          TRAVELER
OUT_EVENT_SUPPRESS_5      ACCELS;LEG;ROUTE;SIGNALS;STOPS;TRIP;TURN;USER;VEHTYPE;VSTYPE;YIELDS
OUT_EVENT_FILTER_5        ANOMALY>2; ANOMALY<8
OUT_EVENT_BEGIN_TIME_5    0
OUT_EVENT_END_TIME_5      86402
```

3. SIGNALIZATION IN THE PORTLAND STUDY

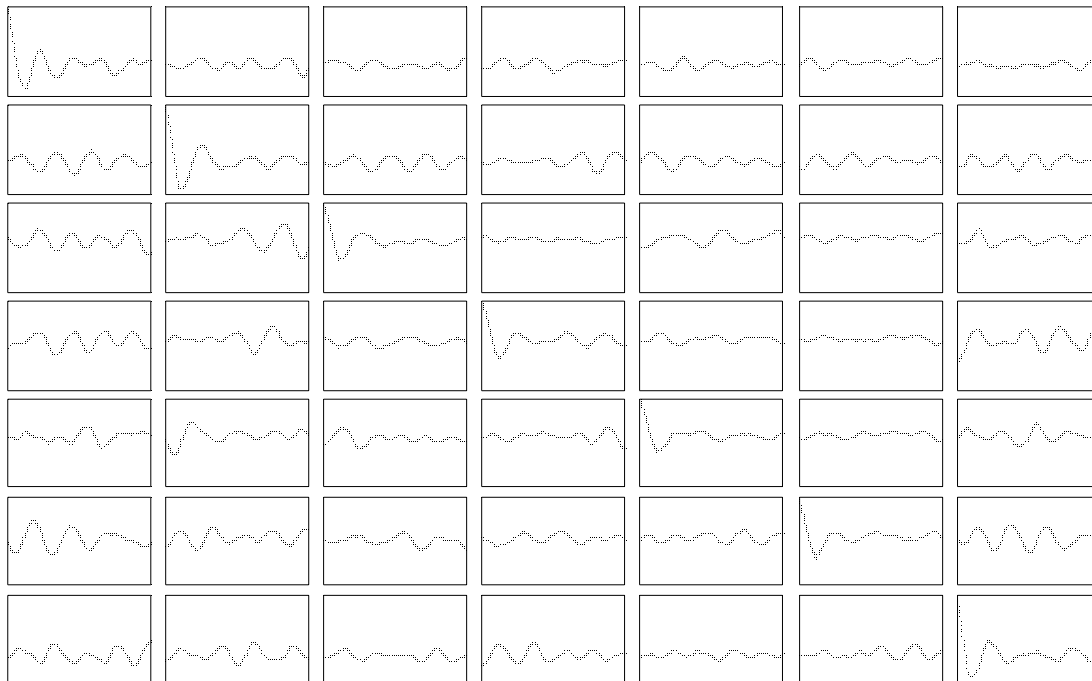
For the Portland, Oregon, case study we used the TRANSIMS traffic control generation program¹ to automatically generate 2086 actuated signals with the parameters very close to the standard set discussed above. We decided not to represent actual signals in the field because of the difficulty of obtaining and processing information describing how the signal controllers in the field had been set.

In order to take advantage of our understanding that well-chosen green times are necessary for coordinated signal operation, we use a signal-generation algorithm where the minimum green length for a phase is based on a weighted sum of the number of incoming lanes participating in the phases, but it is never less than 25 seconds. (A permanent lane has a weight of 1, and a pocket lane has a weight of 1/3.) The total cycle length of at least 60 seconds is apportioned to the phases with constant yellow and red clearance intervals, and minimum green intervals based upon the ratio of the number of incoming lanes in this phase to the total number of lanes. The green extension fraction is set to 60%. The complete set of TRANSIMS configuration settings used to generate the signals is given in Volume Seven (*Appendix: Scripts, Configuration Files, Special Travel Time Functions*), Chapter Eleven (*MS-7*).

We have examined the output of the traffic microsimulation to verify that the set of signal parameters we chose supports adequate throughput of vehicles along key corridors in the network. For example, we studied in detail the corridor containing seven signals between 82nd Avenue between Glissan Street and Division Street (illustrated at the right) between 7:00 a.m. and 7:30 a.m. The plots below show the coordinated operation of signals along this corridor: the rows correspond to the states of the seven signals (ordered from north to south) at a given time, and the columns correspond to their states at a later time represented by the distance along the horizontal axis (0 to 300 seconds); the vertical axis represents the fraction of the time that the later signal is green for southbound traffic given that the earlier signal is green for southbound traffic. As for the five-signal test case discussed above, the coordinated behavior also emerges along the 82nd-Street corridor, but this time in the context of a whole-city simulation. The average travel time for travelers moving south in this corridor is 252 seconds; the average number of seconds stopped in traffic is 66 seconds



¹ C.L. Barrett, R.J. Beckman, K.P. Berkbigler, K.R. Bisset, B.W. Bush, K. Campbell, S. Eubank, K.M. Henson, J.M. Hurford, D.A. Kubicek, M.V. Marathe, P.R. Romero, J.P. Smith, L.L. Smith, P.L. Speckman, P.E. Stretz, G.L. Thayer, E. Van Eeckhout, M.D. Williams, *TRANSIMS: Transportation Analysis Simulation System, Volume 3: Modules*, Los Alamos National Laboratory Report LA-UR-00-1725, 15 June 2001.



3.1 Prospects

We have demonstrated that a single control algorithm with a fixed set of parameters allows for high-performance actuated-signal intersections that are very responsive to varying traffic demand. These can be used in traffic simulations where data on actual controls cannot be easily obtained. Coordination between signals emerges where traffic conditions permit. The signals also perform adequately in real city-wide traffic simulations.

It is possible to continue and extend these calibration studies in several areas: refining the heuristics for choosing parameters, improving our parameter optimization methodology, studying behavior at a variety of intersection types, using larger networks, and looking for the natural emergence of coordination over wide areas. One could also pursue rule-based and pattern-recognition techniques for the automatic generation of controls on traffic networks. The implementation of actuated signals could be enhanced with more complex ring structures, algorithms for specific controllers, coordination of signals (i.e., wide-area control), and ITS technologies.

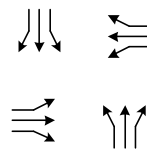
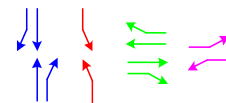
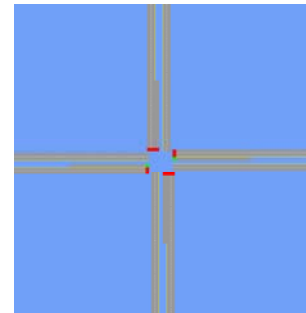
4. GENERIC SIGNAL TUTORIAL

This tutorial outlines how to create generic signals for calibration and testing, and how to run the Traffic Microsimulator to measure their performance. The files for the tutorial reside in the directory *\$TRANSIMS_HOME/data/gensig*.

4.1 Generic Signal Builder

The generic signal builder is a software application for constructing an intersection and vehicular demand for testing the performance of an actuated signal in the TRANSIMS Traffic Microsimulator. It generates the following:

- One intersection with an actuated signal and a specified timing plan.
- Four links in the cardinal directions with a specified length, number of lanes, and speed limit in the north-south and east-west directions.
- Four pocket lanes (one per link) of a specified length for left turns.
- Eight vehicle detectors (two per phase) of a specified length and placement.
- Four phases at the signal: “north-south through,” “north-south left,” “east-west through,” and “east-west left.”
- Plans for vehicles with specified headways for the twelve possible movements at the intersection: “southbound left,” “southbound through,” “southbound right,” “westbound left,” “westbound through,” “westbound right,” “northbound left,” “northbound through,” “northbound right,” “eastbound left,” “eastbound through,” and “eastbound right.”



The TRANSIMS configuration file keys controlling the details of the intersection and demand are given below. The values in the table reflect those used in this tutorial.

Key	Value in Example	Description
CALIB_PLUS_TIME_LIMIT	3600	The number of seconds for which to generate plans.
CALIB_PLUS_SPACING_NORTH_LEFT	0	The spacing [seconds] for vehicles leaving parking and planing to make a left turn movement approaching the intersection from the north.
CALIB_PLUS_SPACING_NORTH_THRU	3	The spacing [seconds] for vehicles leaving parking and planing to make a through movement approaching the intersection from the north.
CALIB_PLUS_SPACING_NORTH_RIGHT	0	The spacing [seconds] for vehicles leaving parking and planing to make a right turn movement approaching the intersection from the north.
CALIB_PLUS_SPACING_EAST_LEFT	18	The spacing [seconds] for vehicles leaving parking and planing to make a left turn movement approaching the intersection from the east.
CALIB_PLUS_SPACING_EAST_THRU	4	The spacing [seconds] for vehicles leaving parking and planing to make a through movement approaching the intersection from the east.
CALIB_PLUS_SPACING_EAST_RIGHT	18	The spacing [seconds] for vehicles leaving parking and planing to make a right turn movement approaching the intersection from the east.
CALIB_PLUS_SPACING_SOUTH_LEFT	24	The spacing [seconds] for vehicles leaving parking and planing to make a left turn movement approaching the intersection from the south.
CALIB_PLUS_SPACING_SOUTH_THRU	7	The spacing [seconds] for vehicles leaving parking and planing to make a through movement approaching the intersection from the south.
CALIB_PLUS_SPACING_SOUTH_RIGHT	21	The spacing [seconds] for vehicles leaving parking and planing to make a right turn movement approaching the intersection from the south.
CALIB_PLUS_SPACING_WEST_LEFT	30	The spacing [seconds] for vehicles leaving parking and planing to make a left turn movement approaching the intersection from the west.
CALIB_PLUS_SPACING_WEST_THRU	24	The spacing [seconds] for vehicles leaving parking and planing to make a through movement approaching the intersection from the west.
CALIB_PLUS_SPACING_WEST_RIGHT	27	The spacing [seconds] for vehicles leaving parking and planing to make a right turn movement approaching the intersection from the west.
CALIB_PLUS_LANES_NS	2	The number of lanes in the north-south direction.
CALIB_PLUS_LANES_EW	2	The number of lanes in the east-west direction.
CALIB_PLUS_SPEED_NS	<default>	The speed limit [meters per second] in the north-south direction. The default is five cells per second.
CALIB_PLUS_SPEED_EW	<default>	The speed limit [meters per second] in the east-west direction. The default is five cells per second.
CALIB_PLUS_GREEN_THRU_NS	20	The number of seconds for greens of the “north-south through” phase. The phase does not exist if this is zero.
CALIB_PLUS_GREEN_LEFT_NS	8	The number of seconds for greens of the “north-south left” phase. The phase does not exist if this is zero.
CALIB_PLUS_GREEN_THRU_EW	20	The number of seconds for greens of the “east-west through” phase. The phase does not exist if this is zero.
CALIB_PLUS_GREEN_LEFT_EW	8	The number of seconds for greens of the “east-west left” phase. The phase does not exist if this is zero.
CALIB_PLUS_GREEN_EXTENSION	0.60	The green extension in terms of the fraction of the initial green. The default is 0.50.
CALIB_PLUS_DETECTOR_LENGTH	37.5	The length [meters] of the detectors. The default is six cells.
CALIB_PLUS_DETECTOR_OFFSET	<default>	The offset [meters] of the detectors, measured from the point of the detector closest to the node to the node. The default is the intersection setback.
CALIB_PLUS_LINK_LENGTH	<default>	The length [meters] of the links. The default is thirty cells plus the intersection setback.
CALIB_PLUS_POCKET_LENGTH	<default>	The length [meters] of pocket lanes. The default is eight cells.
CALIB_PLUS_PARKING_OFFSET	<default>	The offset [meters] of the parking from the edges of the network.
CALIB_PLUS_SETBACK	<default>	The setback [meters] at the intersection. This defaults to an optimal value for the TRANSIMS Visualizer.

The generic signal builder also uses the additional TRANSIMS configuration file keys in the table below—not all of these are required, however.

Key	Required?
NET_DIRECTORY	yes
NET_NODE_TABLE	yes
NET_LINK_TABLE	yes
NET_POCKET_LANE_TABLE	yes
NET_PARKING_TABLE	yes
NET_LANE_CONNECTIVITY_TABLE	yes
NET_UNSIGNALIZED_NODE_TABLE	yes
NET_SIGNALIZED_NODE_TABLE	yes
NET_PHASING_PLAN_TABLE	yes
NET_TIMING_PLAN_TABLE	yes
NET_ACTIVITY_LOCATION_TABLE	yes
NET_PROCESS_LINK_TABLE	yes
NET_STUDY_AREA_LINKS_TABLE	yes
NET_BARRIER_TABLE	yes

Key	Required?
NET_DETECTOR_TABLE	yes
NET_LANE_USE_TABLE	yes
NET_SIGNAL_COORDINATOR_TABLE	yes
NET_SPEED_TABLE	yes
NET_TRANSIT_STOP_TABLE	yes
NET_TURN_PROHIBITION_TABLE	yes
NET_LANE_WIDTH	no
NET_LINK_MEDIAN_HALFWIDTH	no
VEHICLE_FILE	yes
VEHICLE_PROTOTYPE_FILE	yes
PLAN_FILE	yes
OUT_SNAPSHOT_NODES_1	yes
OUT_SNAPSHOT_LINKS_1	yes
CA_CELL_LENGTH	no

The generic signal builder is invoked with a single argument:

```
BuildTestSignal <config-file>
```

where <config-file> is the name of the configuration file containing the keys listed in the two tables above. The directory *\$TRANSIMS_HOME/data/gensig* contains the configuration file *signal.cfg* with the keys used in this tutorial. Running *BuildTestSignal* with this configuration file results in the network files stored in the *network* subdirectory and in the vehicle and plan files stored in the *data* subdirectory.

4.2 Assessing Generic Signal Response

In order to assess the response of the generic signal created by the builder, we run the Traffic Microsimulator and analyze the output. The script *signal.csh* in the directory *\$TRANSIMS_HOME/data/gensig/scripts* runs the does this. After running the Traffic Microsimulator, the *output* subdirectory contains several files of interest:

- *snapshot.veh*: the vehicle snapshot data for the travelers—suitable for viewing in the TRANSIMS Output Visualizer
- *snapshot.sig*: the signal snapshot data for the actuated signal—suitable for viewing in the TRANSIMS Output Visualizer
- *times.trv*: the travel time and inconvenience measure statistics for the travelers—suitable for analysis with the *scripts/counts.awk* script
- *lost.trv*: traveler event data for any travelers that became lost in the simulation—there should be few or none

The script will also produce console output summarizing the response of the signal. Four performance measures are tabulated for each movement at the intersection:

- *flow [veh/hr]*: the flow of vehicles making the movement
- *travel time [s/veh]*: the observed mean and standard deviation for the travel times measured from when a vehicle leaves its parking location on the incoming link to when it enters its parking location on the outgoing link
- *time stopped [s/veh]*: the observed mean and standard deviation for the number of seconds that a vehicle is stopped while it waits to pass through the intersection
- *accelerations from stop [#veh]*: the observed mean and standard deviation for the number of times a vehicle has to accelerate after being stopped on the roadway

For example, in this tutorial the summary should look similar to the following:

```

Movement:                southbound, left
  Flow [veh/hr]:          0

Movement:                southbound, thru
  Flow [veh/hr]:          1177
  Travel Time [s/veh]:     35.3696 (mean), 15.3534 (s.d.)
  Time stopped [s/veh]:    3.19031 (mean), 9.24733 (s.d.)
  Accels. from stop [#veh]: 2.13594 (mean), 0.859248 (s.d.)

Movement:                southbound, right
  Flow [veh/hr]:          0

Movement:                westbound, left
  Flow [veh/hr]:          197
  Travel Time [s/veh]:     86.4162 (mean), 43.3299 (s.d.)
  Time stopped [s/veh]:    11.1015 (mean), 25.0976 (s.d.)
  Accels. from stop [#veh]: 3.30457 (mean), 1.48758 (s.d.)

Movement:                westbound, thru
  Flow [veh/hr]:          894
  Travel Time [s/veh]:     52.4586 (mean), 26.8092 (s.d.)
  Time stopped [s/veh]:    2.84228 (mean), 9.19959 (s.d.)
  Accels. from stop [#veh]: 2.9396 (mean), 1.5013 (s.d.)

Movement:                westbound, right
  Flow [veh/hr]:          199
  Travel Time [s/veh]:     54.5578 (mean), 28.1368 (s.d.)
  Time stopped [s/veh]:    5.09045 (mean), 13.0257 (s.d.)
  Accels. from stop [#veh]: 2.59799 (mean), 1.32543 (s.d.)

Movement:                northbound, left
  Flow [veh/hr]:          139
  Travel Time [s/veh]:     150.367 (mean), 87.6663 (s.d.)
  Time stopped [s/veh]:    17.3813 (mean), 43.9112 (s.d.)
  Accels. from stop [#veh]: 3.54676 (mean), 2.14078 (s.d.)

```

```

Movement:                northbound, thru
  Flow [veh/hr]:          499
  Travel Time [s/veh]:    67.6212 (mean), 55.6657 (s.d.)
  Time stopped [s/veh]:   4.11222 (mean), 10.2163 (s.d.)
  Accels. from stop [#veh]: 3.11824 (mean), 2.12748 (s.d.)

Movement:                northbound, right
  Flow [veh/hr]:          167
  Travel Time [s/veh]:    66.9701 (mean), 57.4042 (s.d.)
  Time stopped [s/veh]:   3.07784 (mean), 8.74332 (s.d.)
  Accels. from stop [#veh]: 2.8024 (mean), 1.66552 (s.d.)

Movement:                eastbound, left
  Flow [veh/hr]:          119
  Travel Time [s/veh]:    61.8739 (mean), 36.7648 (s.d.)
  Time stopped [s/veh]:   18.1008 (mean), 30.0298 (s.d.)
  Accels. from stop [#veh]: 1.72269 (mean), 0.649959 (s.d.)

Movement:                eastbound, thru
  Flow [veh/hr]:          149
  Travel Time [s/veh]:    33.2886 (mean), 19.4514 (s.d.)
  Time stopped [s/veh]:   10.0067 (mean), 15.5748 (s.d.)
  Accels. from stop [#veh]: 1.39597 (mean), 0.555304 (s.d.)

Movement:                eastbound, right
  Flow [veh/hr]:          133
  Travel Time [s/veh]:    33.0752 (mean), 19.1769 (s.d.)
  Time stopped [s/veh]:   10.1955 (mean), 15.1164 (s.d.)
  Accels. from stop [#veh]: 1.43609 (mean), 0.655436 (s.d.)

```

For reference, the demand in this example is given below.

<i>Movement</i>	<i>Demand [veh/hr]</i>
Southbound, left	0
Southbound, through	1200
Southbound, right	0
Westbound, left	200
Westbound, through	900
Westbound, right	200

<i>Movement</i>	<i>Demand [veh/hr]</i>
Northbound, left	150
Northbound, through	514
Northbound, right	171
Eastbound, left	120
Eastbound, through	150
Eastbound, right	133

We can see that nearly all of the vehicles make it through the intersection with the actuated signal.

By editing the CALIB_PLUS_SPACING parameters in the configuration file and then rerunning *BuildTestSignal*, one can evaluate additional demands for the signal under consideration. One can change the actuation algorithm's parameters by editing its configuration parameters:

- *NET_DETECTOR_PRESENCE_SAMPLE_TIME*
- *NET_ACTUATED_ALGORITHM_B_BETA*
- *NET_ACTUATED_ALGORITHM_B_DENSITY_CONST*

- `NET_ACTUATED_ALGORITHM_B_FLOW_CONST`

The signal timing and detector layout may be altered with other `CALIB_PLUS` configuration file keys.

4.3 Troubleshooting

Here are some possible problems that might arise:

- The `$TRANSIMS_HOME` environment variable has not been set.
- You tried to run the Traffic Microsimulator, but do not have write permission to the output directory `$TRANSIMS_HOME/data/gensig/output`.
- You tried to run *BuildTestSignal*, but do not have write permissions to the network, plan, vehicle, or output files specified in the configuration file.
- You have edited the configuration file `$TRANSIMS_HOME/data/gensig/signal.cfg` to change some of the file names or locations, but have not made the corresponding changes in the script `$TRANSIMS_HOME/data/gensig/scripts/signal.csh`.
- You changed the specification for the travel time output in the configuration file `$TRANSIMS_HOME/data/gensig/signal.cfg`—this prevents the script `$TRANSIMS_HOME/data/gensig/scripts/counts.awk` from running because it depends on the details of the output specification.
- The C shell executable is not in `/bin/csh` or the NAWK executable is not in `/bin/nawk` on your system.
- The NAWK executable on your system is a variant of the standard GNU/Linux version.